基于轨迹聚类的航空器轨迹模式挖掘研究

郭 威[a]，唐慧丰[b]

(战略支援部队信息工程大学 a. 研究生院; b. 数据与目标工程学院，郑州 450001)

摘 要：轨迹模式是航空器在某段时间或某个区域内相对稳定的飞行模式，对理解和判断目标在一段时间或一定区域内的行为有着重要的意义。针对目标轨迹的特点，在基于点密度的聚类算法的基础上，设计并实现了一种基于线段密度的轨迹聚类方法。该方法使用最小描述长度原则将目标的历史轨迹分割为若干轨迹段，通过计算轨迹段之间的相似度对飞行轨迹进行聚类，最后运用扫描线算法生成目标的轨迹模式。实验证明，该方法可以较为准确地从大量轨迹数据中发掘出航空器目标的轨迹模式。

关键词：轨迹模式；轨迹聚类；MDL 原则；线段密度；扫描线算法

中图分类号：TP391 doi: 10.19734/j.issn.1001-3695.2018.07.0515

Research on trajectory pattern mining based on trajectory clustering

Guo Wei[a], Tang Huifeng[b]

(*a. Graduate School, b. Data & Target Engineering College, Information Engineering University of Strategic Support Force, Zhengzhou* 450001, *China*)

Abstract: Trajectory mode is a relatively stable flight mode of an aircraft in a certain period of time or a certain area, which is of great significance to understand and judge the behavior of the target in a certain period of time or a certain area. Aiming at the characteristics of target trajectory, a trajectory clustering method based on line segment density was designed and implemented on the basis of point density clustering algorithm. In this method, the historical trajectory of the target was divided into several trajectory segments according to the minimum description length principle, and the flight trajectory was clustered by calculating the similarity between the trajectory segments. Finally, the trajectory pattern of the target was generated by scanning line algorithm. Experiments show that this method can accurately extract the trajectory patterns of aircraft targets from a large number of trajectory data.

Key words: trajectory pattern; trajectory clustering; mdl principle; line segment density; sweep line algorithm

0 引言

人类航空技术的进步推动了航空产业的大发展大繁荣，但日益增多的各类航空器也带来了巨大的监管难题。以飞机为代表的各类航空器在方便人类的同时，对这些机动灵活的空中目标的监管也成了一个棘手的问题。虽然现代社会信息化水平的提高和遥感探测技术的进步使得人类对航空器实时位置的获取和感知变得简单，但由于部分航空器特别是民用航班以外的航空器飞行过程灵活多变，对其行为的分析大部分需要依靠管控人员的知识与经验，难以实现完全的自动化。在飞行数据的规模不断增加和数据的实时分析处理能力严重不足的背景下，如何实现在海量信息中发掘目标的频繁轨迹模式并掌握其飞行特征，对实现包括航空器的飞行管控、飞行轨迹检测与预警、热点区域发掘在内的许多领域有着重要的意义。

轨迹挖掘在早已在气象、生态等领域有着广泛的应用[1]。自然界中，动物的迁徙轨迹、飓风的移动轨迹、洋流的运动轨迹等往往具有一定的规律性，它们的移动路线在一定程度上反映了它们常见的运动趋势。人类受到启发，将这种发现规律的方法应用到更多领域。譬如在军事领域，通过分析群体目标一段时间内的移动路线可以得到目标的活动规律，进而预测目标的作战意图。同样的道理，可以将这类通过分析目标历史轨迹得到目标运动规律的方法应用到航空器轨迹模式的发掘中，用以发现隐藏在海量飞行轨迹信息中的航空器飞行特征与规律。

聚类是目标轨迹模式挖掘的过程中常用的方法之一。在众多的轨迹聚类方法中，本文选取了基于密度的聚类方法DBSCAN（density-based spatial clustering of application with noise）[2]，并结合航空器飞行数据的特点对经典算法加以改进，实现了一种基于线段密度的航空器轨迹聚类算法。该方法通过对目标的历时飞行轨迹数据进行聚类分析，从特定目标的海量历史飞行数据中提取其频繁轨迹模式。在此基础上，可以充分挖掘目标的属性特征和行为特征，进而为实现包括空中目标管控、轨迹异常检测、热点区域发现等应用打下坚实的基础。

1 轨迹数据

航空器的轨迹数据通常指通过卫星、雷达或其它传感器获得的空中目标移动轨迹数据。近年来，移动通信技术和位置感知技术的快速发展使得获取各类移动目标的实时位置信息变得越来越简单，轨迹数据的规模也因此迅速增长。如何在较短时间内从海量的轨迹数据中挖掘出需要的知识和信息，成为了大数据时代数据挖掘领域的一个新挑战。

根据数据挖掘[3]的定义，移动轨迹模式挖掘可以定义为

收稿日期：2018-07-06；**修回日期：**2018-08-23

作者简介：郭威（1993-），男，河南林州人，硕士研究生，主要研究方向为数据挖掘，语言信息处理（guowei1533@qq.com）；唐慧丰（1973-），男，教授，博士，主要研究方向为大数据处理、机器学习.

从海量的、异构的、含噪声的移动轨迹序列中提取潜在的、频繁出现的、具有价值的轨迹序列的过程[4]。移动轨迹数据是指在一定时空环境下，目标运动发生地理位置改变后产生的位置数据。这些数据信息按时间先后的顺序构成了目标的轨迹数据。根据数据采集方式的不同，轨迹数据可以分为基于位置采样的轨迹数据，基于时间采样的位置数据和基于事件触发的轨迹数据[5]。

本文使用的数据是通过多源融合的方式得到的基于时间采样的轨迹数据。具体来讲，飞行数据由一系列按时间排序的多维数据点组成。每个数据点原则上包括点位时间，获取时间经度、纬度、高度、速度、移动方向、获取方式、定位误差以及天气状况等多方面的信息。但是由于采样手段的差异、传感器工作过程和数据传输过程的不稳定性以及其它若干不可控的外部因素，导致实际情况下获取的飞行数据往往是不完整的，部分属性存在缺失或者错误的现象，进而导致最后得到的融合数据中采样时间间隔和采样精度并不统一。因此，为了保证研究结果的可靠性，在进行轨迹聚类前，需要对使用的数据进行预处理，通过计算和先验知识相结合的方式剔除明显错误的数据点，完成清洗降噪等步骤，具体方法在此不再赘述。经过数据预处理过后得到的数据格式如图 1 所示。

编号	时间	经度	纬度	速度	高度
1	t_1	Lng_1	Lat_1	s_1	h_1
2	t_2	Lng_2	Lat_2	s_2	h_2
3	t_3	Lng_3	Lat_3	s_3	h_3
...	...	...	...	...	...

图 1 轨迹数据格式

Fig.1 Format of track data

2 聚类方法选择

聚类是一种常见的技术，是将数据划分成有意义的或有用的组（簇）的过程[6]，被广泛应用于需要处理大量数据的行业与学科。因为航空器在高速飞行过程中会产生大量的轨迹数据，所以在进行航空器轨迹模式发掘的过程中，将每一个目标每一次飞行的轨迹作为单独实体进行处理是不现实的。因此，在实际操作过程中需要对目标的海量轨迹数据进行聚类，获得目标的典型飞行轨迹。经过聚类操作后，可以更容易地从目标的一般轨迹模式中发现目标的飞行规律并提取目标的飞行特征。

聚类的方法[7]有很多种，基本的有基于划分的方法(partitioning method)，基于层次的方法(hierarchical method)，基于密度的方法(density-based method)，基于网格的方法(grid-based method)，基于模型的方法(model-based method)等。在对目标轨迹进行聚类的过程中，可以将整条轨迹进行聚类，也可以选择将轨迹分为若干个子轨迹(sub-trajectory)进行聚类。现有的轨迹聚类算法大多是把目标轨迹作为一个整体进行聚类，分析并得到其共性特点。但是这类算法在使用过程中会遇到许多问题。例如在某些情况下，多条轨迹之间可能存在很多相似的子轨迹，但由于整体不尽相同，所以会将其归为不同类别，导致忽视其共有的特征信息。如图 2 所示，T_1~T_4 是四条不同的轨迹，它们具有共同的部分（图中方框部分），但在整体聚类的思想下，这四条轨迹各不相同，共同的部分被忽略。

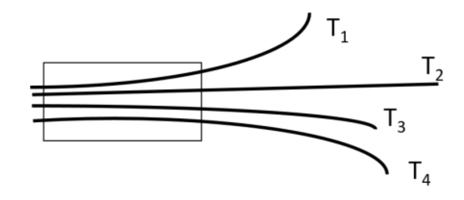


图 2 具有共同子轨迹的实例

Fig.2 Examples of common sub-trajectories

本文涉及的实际应用涉及到特定目标在某一区域的轨迹分析，目的在于发现目标的频繁轨迹模式，通过提取目标轨迹的子轨迹并发现其共同特征具有更加重要的意义，因此本文采用将目标原始轨迹分割为若干子轨迹的方式进行轨迹聚类分析。具体操作上，首先用最小描述长度（minimum description length，MDL）[8]原则对原始轨迹进行分割，得到子轨迹。然后在基于密度的聚类算法 DBSCAN 上针对轨迹聚类的特点加以改进，设计并实现了一种基于线段密度的轨迹聚类算法。这种基于密度的聚类方法可以发现任意形状的类别并对噪声点不敏感，可以根据实际情况生成任意形状的类簇，比较适合线段聚类，更容易从局部片段发现同一目标隐藏在不同飞行轨迹中的特征。最后从得到的轨迹簇中生成目标的轨迹模式。

3 算法描述

传统的基于密度的聚类使用基于中心的方法，即数据集中某一点的密度通过对距该点一定距离内的点计数得到。DBSCAN 是一种简单的、有效的基于密度的聚类算法，用以寻找被低密度区域分离的高密度区域。同时，它解释了基于密度的聚类方法中的许多重要概念，比如邻域，点密度，核心点，直接密度可达，密度可达，密度连接[9]等，这些概念在之后的算法改进和实现的过程中都会提及。

经典的 DBSCAN 算法用于进行点的聚类，实现相对简单。点的密度取决于指定的欧式距离。这种基于中心的方法可以将点分类为核心点（稠密区域的点），边界点（稠密区域边缘的点），噪声点（稀疏区域中的点）。在轨迹聚类中，聚类的对象不是点而是轨迹段。因此，在进行轨迹聚类的过程中，需要对 DBSCAN 算法中的概念进行适当调整，用以适应轨迹聚类的实际情景。在进行轨迹聚类前，引入核心线段、一般线段、噪声线段的定义。

a)核心线段。如果某条线段在给定距离内的线段数量超过阈值，那么该线段属于核心线段。核心线段位于聚类得到的簇内部，线段的邻域由计算距离的函数和指定的距离参数决定。

b)一般线段。一般线段不属于核心线段，它位于某条核心线段的邻域内。一般线段可能落在多个核心线段的邻域内。

c)噪声线段。噪声线段是既非核心线段也非一般线段的其它线段。

给定核心线段、一般线段和噪声线段的定义之后，可以将改进后的基于线段密度的算法描述如下：算法将目标轨迹分割为若干子轨迹，每段子轨迹由若干线段组成。在将多条轨迹划分为若干轨迹段后，然后对这些线段进行聚类；任意两个足够靠近的线段将被归为同一个簇中，某条线段给定领域内的线段超过阈值的被判定为核心线段，一般线段被划分到核心线段所在的簇中，噪声线段被丢弃。该算法的最大优点就在于可以在大规模的轨迹数据中发掘出共同的子轨迹，从而得到相似的路径。

对轨迹的子轨迹进行聚类，就涉及到轨迹的分割问题。

由于目标的轨迹是由一个个按时间顺序的点位构成，最简单的分割方法就是以每两个相邻的点之间的线段作为轨迹的子轨迹。这样分割得到的子轨迹原则上最能代表原始轨迹，但与此同时也带来一些问题：当点位比较密集时会导致子轨迹过短，使得发现轨迹间的共性规律变得更难，而且轨迹划分过多会导致计算量过大。因此，需要找到一个更加合适的轨迹分割方法对原始轨迹进行划分。

最小描述长度是信息论和计算机科学中一个重要概念，简单来讲就是用最少的符号描述最多的内容，符合本文对轨迹进行分割的要求：将一条完整的轨迹在保留其特征的情况下用最少的轨迹段表示。

基于线段密度的轨迹聚类在具体操作时分为分割、聚类和重组三个阶段。在分割步骤中使用 MDL 原则对原始目标轨迹进行分段表示；在聚类阶段，采用基于密度的（density-based）线段聚类算法[10]对相似的目标轨迹分段进行聚类；在重组阶段，从聚类得到的轨迹段集合中生成目标的轨迹模式。

假设所有 n 条轨迹位于同一集合 TRA 中，TRA={T_1，T_2，T_3，…，T_n}。每条轨迹由一系列多维的数据点 p 构成。假设轨迹 T_i 由 leni 个点构成，则轨迹 T_i={p_1，p_2，p_3，...，p_{leni}}，轨迹 $p_{c1}p_{c2}\ldots p_{ck}$(1<c1<c2<ck<leni)被称为 T_i 的子轨迹。在算法实现过程中，每条轨迹首先被分割为一系列可以代表该条轨迹的轨迹段集合 C={C_1，C_2，…，C_n}，其中 C_i（0<i<n）为轨迹 T_i 的分割后的线段。线段 p_ip_j（i<j）是轨迹分割后的某一线段，线段之间的距离如果小于某一阈值，则会被归到同一类簇中。每条轨迹段随着算法的推进都会被判定为核心线段、一般线段或噪声线段。最后从聚类得到的轨迹簇中提取目标的代表性轨迹（representative trajectory）并生成目标轨迹模式。代表性轨迹由一系列的点位构成，是表征目标主要飞行特征的模拟轨迹。完整算法的伪代码如下所示。

输入：目标原始轨迹集合 TRA={T_1，T_2，T_3，…，T_n}

输出：目标规律轨迹集合

算法：

```
step1
  foreach T in TRA
    execute sub-trajectory using MDL principle
    get a line segment set D
step2
  foreach line segment L in D
    execute line segment cluster using DBSCAN
step3
  execute representative trajectory
```

4　算法原理

针对航空器在飞行过程中的轨迹特点，论文对在 DBSCAN 算法的基础上设计并实现了一种基于线段密度的轨迹聚类算法。该算法分为三个阶段：第一阶段使用 MDL 原则对原始轨迹进行分割，使得分割后的轨迹可以在保证不失真的情况下尽可能地简洁，简化了计算过程，减少了计算时间和空间的消耗；第二阶段是设计并使用了一种基于组合距离的线段相似度判别函数对轨迹段进行聚类，把相似的轨迹段合并为同一类簇；第三阶段是从各个轨迹簇中生成代表性轨迹作为目标的轨迹模式。

4.1　基于 MDL 原则的轨迹分割与表示

本节介绍如何在尽可能保持轨迹原貌的前提下对轨迹进行分割。在对轨迹进行分割时，最理想的状态就是找到轨迹特征变化大的点位，称之为特征点(characteristic point)。对于轨迹 $T_i=p_1p_2p_3\ldots p_{leni}$，选取了一系列特征点{$p_{c1}$，$p_{c2}$，$p_{c3}$，…，$p_{cn}$}（$c_1<c_2<c_3<\ldots<c_n$）对轨迹 T_i 进行分割，分割后每两个相邻的特征点连接为一条线段，所以轨迹 T_i 被分割为 n-1 条线段。图 3 展示了轨迹分割的过程，其中实线部分是原始轨迹，虚线部分是分割后得到的特征轨迹，p_1、p_3、p_5 为特征点。

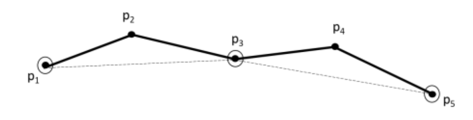


图 3　轨迹分割的过程

Fig.3　The process of trajectory segmentation

在用以上方法对轨迹进行分割的过程中，要考虑两方面的因素：准确和简明。准确就是要求分割后的轨迹要和原始轨迹的差距尽可能地小；简明就是要求轨迹分割后的轨迹段数目应尽可能地少，即要求在对轨迹进行分割的过程中尽可能选取轨迹特征变化明显的点位。但与此同时，特征点位选取数量较少的情况下会使得分割后轨迹的准确性无法得到保证。由于轨迹分割很难同时满足准确和简明两个要求，因此要在二者之间寻求一个最佳的平衡，最大限度地满足准确和简明的要求。针对以上要求，论文采用在信息论中被广泛应用的最小描述长度（MDL）原则对目标原始轨迹进行分割。

在 MDL 原则中，最小长度包含两部分，L(H)和 L(D|H)。其中 L(H)是假设所占用的信息长度，L(D|H)是在假设基础上表示目标所用信息长度。在把 MDL 原则应用到轨迹分割时，L(H)和 L(D|H)代表的内容也有相应的改变。L(H)代表按照特征点对原始轨迹分割后得到的轨迹段长度之和，L(D|H)代表了原始轨迹和分割后轨迹段之间的差异程度，这种差异程度通过距离函数来确定。本文算法表示差异程度所使用的距离函数由原始轨迹和分割后轨迹段之间的三种距离之和来表示。这三种距离分别是垂直距离 $d_{\perp}$，角度距离 d_{θ} 和平行距离 $d_{\parallel}$，具体定义及计算方法将在下节详细解释。有关 L(H)和 L(D|H)的实例如图 4 所示，其中 p_1、p_4 为假设的特征点。

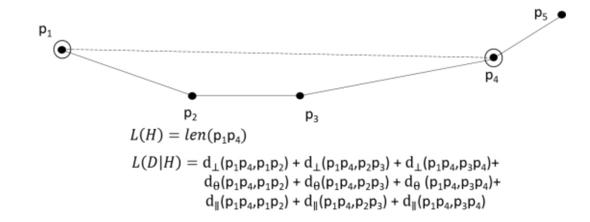


图 4　L(H)和 L(D|H)的实例

Fig.4　Examples of L (H) and L (D|H)

从有关公式的定义可以看出，L(H)代表了分割的简明性，L(D|H)代表了准确性。在特征点选择时，L(H)随着分割的轨迹段的数量增加而增大，L(D|H)随着分割后的轨迹偏离原始轨迹的程度而增大。

在寻找轨迹分割点（特征点）时，为了简化计算，采用局部最优代替全局最优的方式对分割点进行选择。假设当 p_i 为特征点时，用 $L_{par(i)}$ 代表 L(H)和 L(D|H)之和；当 p_i 为普通数据点时，$L_{nopar(i)}$ 代表轨迹的原始长度。在判断 p_i 是否为特征点时，将 $L_{par(i)}$ 和 $L_{nopar(i)}$ 进行比较，当 $L_{par(i)} \le L_{nopar(i)}$ 时，可以把 p_i 作为分割点。并且当 L(H)和 L(D|H)的和值最小时，轨迹分割的效果最优。算法的伪代码如下所示。

输入：目标原始轨迹 T={$p_1,p_2,p_3,p_4,\ldots,p_{leni}$}

输出：轨迹的分割点集合 C

算法：

```
add p1 into the C
start=1,length=1
while(start+length≤leni)
  i=start+length
  if (Lpar(start,i)<Lnopar(start,i))
    Add pi into C
    startIndex=i
    length=1
  else
    length++
add pleni into C
```

4.2　基于组合距离的相似度判别函数

在对分割后的轨迹段进行聚类之前，首先对聚类过程中计算相似度用到的几个距离函数进行简要介绍。在基于组合距离的相似度判别函数由三个距离组合得到：垂直距离、平行距离、角度距离。

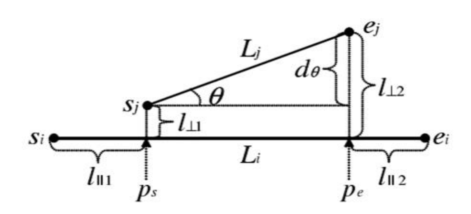


图 5　垂直距离、平行距离、角度距离展示图

Fig.5　Display of vertical distance, parallel distance and angle distance

如图 5 所示，假设有两个 d 维的轨迹线段 $L_i=s_ie_i$，$L_j=s_je_j$，s_i、e_i、s_j、e_j 分别是两条线段的首尾点，p_s 和 p_e 分别为 s_j 和 e_j 在 L_i 上的投影点。l_1 是 s_j 和 p_s 之间的欧氏距离，l_2 是 e_j 和 p_e 之间的欧氏距离，则 L_i 和 L_j 之间垂直距离的定义如式（1）所示；假设 p_s 和 p_e 分别为 s_j 和 e_j 在 L_i 上的投影点，l_1 表示 p_s 到 s_i 的距离，l_2 表示 p_e 到 s_i 的距离，则 L_i 和 L_j 的平行距离的定义如式（2）所示；假设$|L_i|$表示 L_i 的长度，$|L_j|$表示 L_j 的长度，θ 表示 L_i 和 L_j 之间的夹角，可以通过向量的方式计算得到，则 L_i 和 L_j 之间的角度距离定义如式（3）所示。

$$d_\perp\left(L_i,L_j\right)=\frac{l_{\perp 1}^2+l_{\perp 2}^2}{l_{\perp 1}+l_{\perp 2}} \tag{1}$$

$$d_\parallel\left(L_i,L_j\right)=\mathrm{Min}\left(l_{\parallel 1},l_{\parallel 2}\right) \tag{2}$$

$$d_\theta\left(L_i,L_j\right)=\begin{cases}\|L_j\|\times\sin\theta, 0°\le\theta\le 90° \\ \|L_j\|, 90°\le\theta\le 180°\end{cases} \tag{3}$$

垂直距离、平行距离和角度距离的定义与计算方法介绍完毕。两条线段之间相似性的判别函数可由三种距离组合得到，假设 dist（L_i，L_j）表示 L_i 和 L_j 之间的距离判别函数，则其定义如式（4）所示。dist（L_i，L_j）的值越大，L_i 和 L_j 的相似度越低。

$$dist\left(L_i,\ L_j\right)=d_\perp\left(L_i,\ L_j\right)+d_\parallel\left(L_i,\ L_j\right)+d_\theta\left(L_i,\ L_j\right) \tag{4}$$

4.3　基于线段密度的轨迹聚类

为了描述方便，除了之前提到的核心线段、边界线段、噪声线段外，还需引入三个轨迹聚类过程中用到的定义。假设 D 代表所有分割后的轨迹段集合，L_i、L_j 代表 D 中的任意一条轨迹段。

a)ε-邻域。轨迹段在给定距离 ε 以内的区域称为该轨迹段的 ε-邻域，用符号表示为 N_ε。

b) *MinLns*。判定轨迹段是否为核心轨迹的周围轨迹段数量阈值。如果一个轨迹段的 ε-邻域内的轨迹段数量大于或等于 *MinLns*，则该轨迹段为核心轨迹段。

c)直接密度可达。在集合 D 中， L_i 在 L_j 的ε-邻域内并且 L_j 为核心轨迹段，那么 L_i 到 L_j 直接密度可达。

d)密度可达。在集合 D 中，存在轨迹段 L_1，L_2，…，L_n，$L_i=L_1$，$L_j=L_n$，如果 L_k 到 L_{k-1}（$1<k\le n$）直接密度可达，那么轨迹段 L_i 到 L_j 密度可达。

e)密度相连。在集合 D 中，如果轨迹段 L_k 到 L_i 和 L_j 都是密度可达的，那么称 L_i 和 L_j 密度相连。

基于线段密度的轨迹聚类算法的目的就是在轨迹集合 D 中寻找密度相连轨迹段的最大集合 O，实现过程中需要设置三个参数：邻近阈值 ε 和数量阈值 $MinLns_1$ 和 $MinLns_2$。算法开始前，所有被分割后的轨迹段被标记为未识别的轨迹段，随着算法的推进，这些轨迹段都会被标记为不同的类型，划分到某一簇或者被标记为噪声轨迹。为了便于描述，可将算法可以分为三个阶段。

a)计算每一条未被标记的轨迹段的 ε-邻域。如果轨迹段 L 被标记为核心轨迹段，算法就会执行第二部分，计算并得到其 ε-邻域。

b)计算该区域内所有轨迹段的直接密度可达轨迹段并将其加入到由核心轨迹段所形成的簇中。如果一个新加入的轨迹段未被判定是否为核心轨迹，将会被加入到队列 Q 中等待判断。判定过程中用到参数 $MinLns_1$。

c)计算每个簇中轨迹段所在原始轨迹的数目，如果小于阈值 $MinLns_2$ 则将其过滤。

算法的伪代码如下所示。

输入：轨迹段集合 D={L_1,L_2…L_n}；参数 ε 和参数 $MinLns_1$, $MinLns_2$。

输出：轨迹集合 O={C_1,C_2…C_n}。

算法：

```
step1
mark all the line segments in D as unclassified
foreach L in D
  if（L is unclassified）
    compute Nε(L)
    if(|Nε(L)|≥MinLns1)
      assign clusterID to ∀X∈Nε(L)
      insert Nε(L)-{L} into queue Q
step2
while(Q≠Ø)
  Compute Nε(M)//M is one of a segment in Q
  if(|Nε(M)|≥MinLns1)
    foreach Y in Nε(M)
      if (Y is unclassified)
        assign clusterID to Y
      else
        mark M as noise
      Remove M from the queue Q
      clusterID ++
  else
    mark L as noise
step3
foreach C in O
  if |C|<MinLns2
    Remove C from the set O of clusters
```

4.4　特征轨迹生成

在对轨迹进行聚类过后会得到若干轨迹簇，算法的最后一步就是从轨迹簇中生成可以代表总体运动趋势的特征轨

迹。特征轨迹由一系列的点位组成，这些点位通过扫描线算法[11]得到。

扫描线算法的流程如下：对于在算法上一步中得到的轨迹段集合，在轨迹所在的坐标系内，生成一条和集合内轨迹趋势垂直的线段，称之为扫描线；通过计算每条轨迹段和扫描线的交点个数判断是否在相应位置生成特征轨迹的点位，如果相交的点位超过某一设定好的阈值，那么会计算这些交点的坐标均值作为特征轨迹点位的坐标；最后将这些特征轨迹点位连接起来构成该轨迹集合的特征轨迹。算法的最后，将所有轨迹段集合的特征轨迹合并，构成目标的轨迹模式。算法的实例如图 6 所示，图中黑色实线部分是轨迹段集合，虚线为扫描线，扫描线按箭头方向移动，交点阈值假定为 3。经过计算生成的特征点用黑色实心点表示，将这些特征点相连生成的轨迹作为集合的特征轨迹。

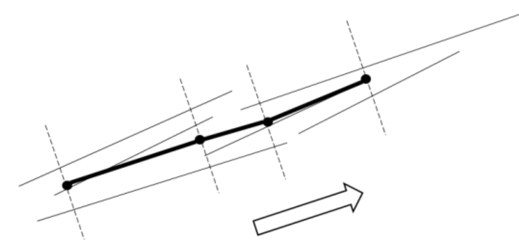


图 6　特征轨迹生成实例

Fig.6　Example of feature path generation

5　实验仿真及结果分析

本节通过进行仿真实验验证算法的有效性。首先介绍了实验环境和实验数据，接着把经过预处理后的轨迹数据和参数作为输入进行运算，最后把得到结果后将其用可视化的技术呈现。

5.1　实验环境

硬件环境：Windows7 操作系统，Intel Core-i7 处理器（3.6 GHz），内存 8 GB。

编译语言：Java。

编译环境：Eclipse Neon，JDK 1.8。

5.2　数据选择与参数设置

实验验证环节使用了 A 型飞机（注：由于实验数据涉密原因，目标名称、目标数据和实验结果在展示时均进行了处理）在 2017 年第一季度的飞行轨迹数据。该数据集包含了 157 条轨迹，共 23 035 个数据点。实验之前，已经将数据进行过清洗，最大限度地保证了数据的完整性和可靠性。

实验过程中需要对参数 ε、$MinLns_1$ 和 $MinLns_2$ 进行设置。在前期程序设计过程中的小规模数据测试时发现，三个参数的取值大小对于实验结果有较大的影响。当 ε 一定时，$MinLns_1$ 的取值和聚类生成的轨迹簇数目的成负相关；当 $MinLns_1$ 一定时，ε 的取值和聚类生成的轨迹簇数目的成正相关。$MinLns_2$ 的大小也会影响到最后生成的轨迹簇数目。由于轨迹簇过多或过少都难以客观准确地得出目标的特征轨迹，因此在设置参数时需要根据实际情况并结合经验知识合理地调整参数的大小。

在基于密度的聚类方法中对于参数的优化算法有许多种，论文选取了理解和实现相对容易的比较误差平方和（Sum of Squared Error ,SSE）的方法[12]，在保证误差平方和相对较小的情况下结合人工经验选取参数，使得选取的参数在聚类过程中生成最能代表目标飞行模式的特征轨迹。通过多次试验和优化，最终确定 ε 的取值为 0.45，$MinLns_1$ 和 $MinLns_2$ 的取值为均为 10。

5.3　程序运行及结果分析

在这里不再使用平行距离因为在一般情况下航空器的原始轨迹和分段后轨迹比较相似，为了减少轨迹较长、数据量较大时计算量过大的情况，忽略掉平行距离。

由于实验数据是位置信息数据，为了更直观地展示实验过程，实验环节采用了可视化的技术将数据根据其地理位置在地图上显示出来。数据可视化过程中，论文使用了基于 Python 语言的开源工具包 Matplotlab 和 Basemap[13]。

图 7 展示了数据预处理前后目标轨迹的示意图。通过对比可以明显地看出，在进行数据预处理之前的目标轨迹中存在着大量错误点。图中的错误点的原因是由于采集手段的不同导致融合过后的数据不统一，在数据预处理过程中对明显错误的数据点予以剔除。因此，数据预处理步骤十分必要，否则会影响到之后目标轨迹的聚类精度。

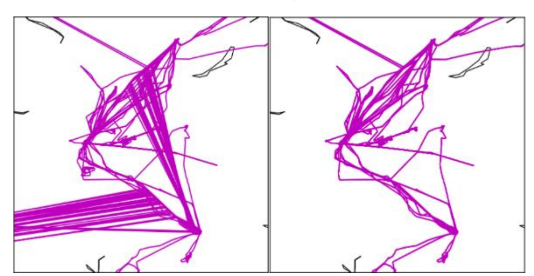


图 7 数据预处理前后对比图

Fig.7　Comparison before and after data preprocessing

图 8 展示了目标原始轨迹经过完整的轨迹聚类步骤之后得到的目标轨迹模式。从图中可以看出，算法成功地从历史轨迹中提取出三条目标的轨迹模式。实验得到的结果经过领域专家人工判证，符合领域的先验知识，较为准确地刻画出 A 型飞机的轨迹模式和活动规律，达到了实验预期效果，证明了算法的合理性。

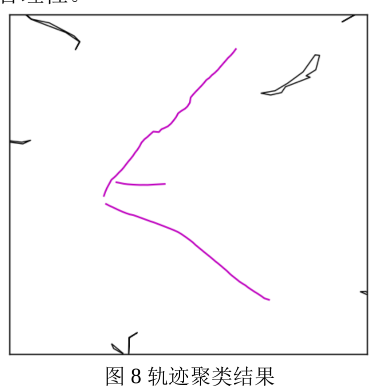


图 8 轨迹聚类结果

Fig.8　Track clustering results

6　结束语

本文设计并实现了一种基于线段密度的轨迹聚类方法用于航空器轨迹模式的发现。该方法的实现过程分为原始轨迹分割、轨迹段聚类和特征轨迹生成三个阶段。在轨迹划分阶段，使用最小描述长度准则将目标的原始轨迹分割为轨迹段；轨迹聚类阶段，使用改进的基于密度的算法对划分后的轨迹段进行聚类；特征轨迹生成阶段，使用扫描线算法提取出特征轨迹作为目标的轨迹模式。最后，通过实验仿真验证了方法的合理性和实用性。除了用于航空器轨迹模式的挖掘之外，针对特定领域进行相应改进，该方法在海空目标的管控、目标的轨迹预测、热点区域的发现等方面同样具有重要的意义和价值。下一步的工作将致力于针对不同情况下算法参数的优化和通过运动趋势对目标的轨迹进行预测。

参考文献：

[1] 袁冠，夏士雄，张磊. 基于结构相似度的轨迹聚类算法 [J]. 通信学报，2011, 32 (9): 103-110. (Yuan Guan, Xia Shixiong, Zhang Lei. Trajectory clustering algorithm based on structural similarity [J]. Journal of Communications, 2011, 32 (9): 103-110.)

[2] Han Jiawei, Kamber M, Pei Jian. 数据挖掘：概念与技术 [M]. 范明，孟小峰，译. 3 版. 北京：机械工业出版社，2012: 1-54. (Han Jiawei, Kamber M, Pei Jian. Data mining: concept and technology [M]. Fan Ming, Meng Xiaofeng, Trans. 3rd ed. Beijing: Machinery Industry Press, 2012: 1-54.)

[3] Apostolakis J. An introduction to data mining [M]// Data Mining in Crystallography. Berlin: Springer, 2009: 1-35.

[4] 刘大有，陈慧灵，齐红，等. 时空数据挖掘研究进展 [J]. 计算机研究与发展，2013, 50(2): 225-239. (Liu Dayou, Chen Huiling, Qi Hong, *et al.* Advances in spatio-temporal data mining [J]. Computer Research and Development, 2013, 50 (2): 225-239.)

[5] 安建瑞. 基于 MapReduce 的用户移动轨迹序列模式挖掘算法研究 [D]. 淄博：山东理工大学，2016. (An Jianrui. MapReduce-based algorithm for mining user motion trajectory sequence patterns [D]. Zibo: Shandong University of Technology, 2016.)

[6] Pang-Ning Tan, Michael Steinbach, Vipin Kumar. Introduction to Data Mining [M]. Boston:Addison-Wesley Longman Publishing Co. Inc., 2005.

[7] 虞倩倩. 基于数据划分的 DBSCAN 算法研究 [D]. 无锡：江南大学，2013. (Yu Qianqian. Research on DBSCAN algorithm based on data partition [D]. Wuxi: Jiangnan University, 2013.)

[8] Lee J G, Han Jiawei, Whang K Y. Trajectory clustering: a partition-and-group framework [C]//Proc of ACM SIGMOD International Conference on Management of Data. 2007: 593-604.

[9] 邢冬丽，赵美红，陈文成. 基于密度的 DBSCAN 算法 [J]. 计算机工程与应用,2007,43(20): 216-221. (Xing Dongli, Zhao Meihong, Chen Wencheng. Density-based DBSCAN algorithm [J]. Computer Engineering and Applications, 2007, 43 (20): 216-221.)

[10] Ankerst M, Breunig M M, Kriegel H P. OPTICS: ordering points to identify the clustering structure [J]. ACM SIGMOD Record, 1999, 28 (2): 49-60.

[11] Shamos M I, Hoey D. Geometric intersection problems [C]//Proc of Symposium on Foundations of Computer Science.Piscataway,NJ: IEEE Press,1976: 208-215.

[12] Han J, Kamber M. Data mining: concepts and techniques [J]. Data Mining Concepts Models Methods & Algorithms, 2006, 5 (4): 1-18.

[13] 李磊，郑锦娜，王心华. 数字填图地理底图转换与制作方法研究 [J]. 地质调查与研究，2013,36 (4):318-323.(Li Lei,Zheng Jinna, Wang Xinhua. Study on the conversion and making methods of digital mapping geographic map [J]. Geological Survey and Research,2013,36 (4): 318-323.)